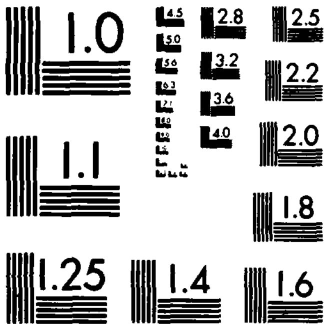
AD-A137 124 PROGRAMMING PRODUCTIVITY ENHANCEMENT BY THE USE OF APPLICATION GENERATORS(U) UNIVERSITY OF SOUTHERN CALIFORNI LOS ANGELES DEPT OF COMPUTE.. HOROWITZ 1/1

UNCLASSIFIED NOV 83 AFOSR-TR-83-1331 AFOSR-82-0232 F/G 9/2 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

6a. NAME OF PERFORMING ORGANIZATION University of Southern California	6b. OFFICE SYMBOL *(If applicable)*	7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research
6c. ADDRESS *(City, State and ZIP Code)* Computer Science Department, University Park, Los Angeles CA 90089-0782		7b. ADDRESS *(City, State and ZIP Code)* Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR	8b. OFFICE SYMBOL *(If applicable)* NM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-82-0232
8c. ADDRESS *(City, State and ZIP Code)* Bolling AFB DC 20332		10. SOURCE OF FUNDING NOS.
11. TITLE *(Include Security Classification)* SEE REMARKS ON BACK		

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2304	A2	

12. PERSONAL AUTHOR(S)
Ellis Horowitz

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Yr., Mo., Day)*	15. PAGE COUNT
Interim	FROM 6/1/83 TO 11/1/83	NOV 83	4

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUB. GR.	

ITEM #11, TITLE:
PROGRAMMING PRODUCTIVITY ENHANCEMENT BY THE USE OF APPLICATION GENERATORS

ITEM #19, CONTINUED: The main activity during the previous nine months has been to see if the investigators could design these features into a general purpose programming language. They decided to use Ada as the starting point. ~~To begin with,~~ they needed to design an extension of Ada that permits the language to interface with a database management system. They concluded that this interface should not merely be a set of remote procedure calls, but a true extension of the language. This implied that they had to extend the type facility and provide new operators, while preserving the design principles of the language. This is the logical first step towards their goal of incorporating true application generator features into a conventional programming language.

This report summarizes the work in this area during this period.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

AFOSR-TR- 83-1331

Research Progress and Forecast Report for Grant No. AFOSR-82-0232 Programming Productivity Enhancement by the Use of Application Generators

June 1, 1983 - Nov. 1, 1983

Principal Investigator
Dr. Ellis Horowitz
Computer Science Department
University of Southern California
Los Angeles, Ca. 90089
213-743-6453

submitted to
Dr. Robert Buchal
AFOSR/NM
Building 410
Bolling AFB, DC 20332
202-767-5025

1. Research Progress to Date

This research was initiated in June, 1982. The first seven months of progress was detailed in the previous annual report. Summarizing briefly here, the early work began with an investigation of commercially available application generators. This was undertaken because of a belief that such systems provide a major increase in programming productivity, at least for a narrow range of "edp" applications. The plan called for investigating systems such as RAMIS, NOMAD and FOCUS with the goal of determining what features contributed to this improvement. We were successful in that we isolated what we believe to be the major features that contribute to increased programming productivity, namely an application generator's built-in interface to a database management system, its non-procedural programming language constructs, and the high-level operators for specific operations. This study and analysis was summarized in the paper *An Analysis of Application Generators*.

Our main activity during the previous nine months has been to see if we could design these features into a general purpose programming language. We decided to use Ada as the starting point. To begin, we needed to design an extension of Ada that permits the language to interface with a database management system. We concluded that this interface should not merely be a set of remote procedure calls, but a true extension of the language. This implied that we had to extend the type facility and provide new operators, while preserving the design principles of the language. This is the logical first step towards our goal of incorporating true application generator features into a conventional programming language.

In contrast to others, the database language extensions we have designed are based upon the relational data model. The system is interfaced to a relational database management system via a new Ada type *relation*. The language includes basic operations on relations, similar to those commonly available in database query languages, such as retrieval of data, updating of tuples and high-level operators to combine relations to form new ones. We have shown how Ada exception handling is naturally extended to allow integrity control of the relations. We also have devised language features that enable the sharing of objects among several users. The entire extension of Ada is reported on in our paper entitled *AdaRel: A Relational Extension of Ada*.

The second major activity during this period has been the investigation of Integrated Office Information Systems. It is clear to us that the nature of programming is radically changing. By studying the programming needs of offices, which are often data intensive, we hope to be led towards a new paradigm for programming that icorporates the virtues of application generators. From this study we have concluded that our starting point should be the "form", something we all deal with often in our everyday affairs. An integrated office information system based upon forms offers a non-procedural way of programming office applications and so it is properly viewed as a manifestation of an application generator. We have now defined the basic properties of a form and the operations that must be supported by a forms system if a system based upon it will be capable of describing the complete range of programming tasks. These include 1. form template definition; 2. form template instantiation; 3. specifying actions on form instances such as mailing, copying, saving and triggering; 4. validation of forms; and 5. storage and retrieval of forms and their contents. This work has been summarized in our paper *The Design of Office Information Systems.*

Status of the Related Computer Activities

The SUN computer workstations have arrived in this period. They are currently having Berkeley Unix version 4.2 installed and the Ethernet cabling is being run to all the offices on the second floor of this building. Our computing center anticipates a fully operational network by Jan. 1, 1983. The computing center will also be purchasing a second DEC VAX running UNIX. We expect to use the SUNs for developing and prototyping our designs in the upcoming period.

Goals and Timetable for the Period 11/1/83 - 5/31/83

Right now all of our work has focussed on the Ada language extension. Once that is complete, we plan to consider applying the techniques to the VAX-UNIX environment. We plan to do this by designing application generator features that will be processed into C programs. These programs will interface with a relational database management system, most likely Unify. We had to reject the use of Ingress because it is too big and too slow. We examined and eventually rejected TROLL because of its low-level nature and the fact that the higher-level language that it was supposed to interface with, called PLAIN, was never implemented.

Our next immediate step will be to take AdaRel and extend it further so that it includes application generator features. We are studying two major areas: report generation and input/output. For example, we intend to design a set of language constructs that facilitate the design of interactive programs whose purpose is to display and collect data from/to a database management system. Such an interactive dialog consists of a set of screens that are displayed to the user in sequence. The screens display information and request further information form the user.

4. Current Status of Personnel and Equipment In addition to the Principal Investigator, Ellis Horowitz, there are two Computer Science graduate students who are being supported on this grant, Alfons Kemper and Balaji Narasimhan. Mr. Kemper has been focussing on the application generator in Ada work. Mr. Narasimhan has been investigating language designs for forms processing. Both are working towards their Ph.D. degrees and their anticipated graduation is June, 1985.

5. Publications Status

1. "An Analysis of Application Generators", Technical Report TR-83-208, Computer Science, USC, March 1983, submitted to *IEEE Trans. on Software Engineering.*

2. "AdaRel: A Relational Extension of Ada", Technical REport TR-82-309, Computer Science, USC, October, 1983, submitted to *ACM Trans. on Programming Languages.*

3. "The Design of Office Information Systems", Technical Report TR-83-320, Computer Science, USC, October, 1983, to be submitted to *ACM Trans. on Office Systems.*